SYSTEM AND METHOD FOR MATCHING STORAGE DEVICE QUEUE DEPTH TO SERVER COMMAND QUEUE DEPTH

Inventors: Jacob Cherian
12345 Lamplight Village Ave. Apt 1524
Austin, Texas 78758

Thomas J. Kocis
11505 Citrus Cove
Austin, Texas 78750

Assignee: DELL PRODUCTS, L.P.

BAKER BOTTS L.L.P.
One Shell Plaza
910 Louisiana
Houston, Texas 77002-4995

Attorney's Docket: 016295.0641
DC-02789

SYSTEM AND METHOD FOR MATCHING STORAGE DEVICE QUEUE DEPTH TO SERVER COMMAND QUEUE DEPTH

TECHNICAL FIELD

The present disclosure relates generally to the field of storage area networks and, more particularly, to a system and method for matching in a storage area network the queue depth of the various storage devices to the execution throttle of the servers of the storage area network.

BACKGROUND

A storage area network (SAN) may be used to provide centralized data sharing, data backup, and storage management. A storage area network is a high-speed network of shared storage devices. Elements of a SAN include servers, switches, storage controllers, and storage devices. A storage device is any device that principally contains a single disk or multiple disks for storing data for a computer system or computer network. Each server is usually connected to the network by a host bus adapter (HBA) and will include an HBA device driver, which is the software driver for the HBA. The collection of storage devices is sometimes referred to as a storage pool. The storage devices in a SAN can be collocated, which allows for easier maintenance and easier expandability of the storage pool. The network architecture of most SANs is such that all of the storage devices in the storage pool are potentially available to all the servers that are coupled to the SAN. Additional storage devices can be easily added to the storage pool, and these new storage devices will also be accessible from any server on the SAN.

In a computer network that includes a SAN, the server can act as a pathway or transfer agent between the end user and the stored data. Network servers can access a SAN using the Fibre Channel protocol, taking advantage of the ability of a Fibre Channel fabric to serve as a common physical layer for the transport of multiple upper-layer protocols, such as SCSI, IP, and HIPPI, among other examples. With respect to storage, any element of storage on a SAN may be assigned

10970604.120501

its own SCSI logical unit number (LUN). The LUN address of each storage element is used to identify the storage element within the SAN. Each storage element and its associated LUN are assigned to one or more of the servers of the SAN. Following this assignment, each server will have logical ownership of one or more LUNs, allowing data generated by the server to be stored in the storage devices corresponding to the LUNs owned by the server.

Along with its advantages, the SAN environment also has additional complexities. One of the complexities relates to the relationship between the execution throttle levels of servers and command queue depth settings of storage controllers in a SAN. As noted above, a SAN may be used to provide access to multiple storage devices from multiple host servers. In a SAN, the number of hosts that each storage controller can support is described as the fan out for the storage controller. The fan out value for a storage controller is the number of servers that can access its storage resources. The command queue depth of the storage controller is the maximum number of input/output (I/O) commands that the device can queue up for processing, past which the commands are either dropped or returned with a busy status. Dropping commands or returning a busy status results in a degradation of the performance of the overall system. The maximum number of I/O commands that a server can have outstanding is generally referred to as the server's execution throttle. The execution throttle is typically controlled by configuration settings in the HBA device driver. A SAN will generally operate most efficiently when the command queue of each storage controller of the network is at or near capacity without ever having exceeded capacity.

A difficulty with the selection of execution throttles for the servers of the SAN concerns the possibility that the execution throttle of a server will be set too high or too low. When an execution throttle value is set too high, that is, when the sum of execution throttles on host servers that own LUNs on a particular storage device exceeds the command queue depth for that storage device, the total I/O demand from the servers is too high as compared to the command capacity, as measured by the command queue depth, of the storage controllers of the SAN, resulting in dropped commands or busy signals returned by the storage controllers. In addition, when an execution throttle value is set too low on a particular server, such as when a low setting is desired in order to insure that the I/O demand from the servers does not exceed the command capacity of the coupled

storage controllers, the performance of the server may be severely affected by the inability to process commands fast enough to satisfy the demands of the operating system and application programs that generate the commands.

For many SANs and SAN administrators, a common method for selecting the execution throttle levels of the servers of the SAN is to set the execution throttle for each host to a value that is determined at setup by dividing the smallest command queue depth value of all of the storage controllers on the SAN by the number of servers. This method, however, does not take into account the possibility of varying command queue depths among the several storage controllers on the SAN. As a result, all storage controllers, including those having the capacity for higher command queue depths, are treated as though they have the capacity of the storage controller with the smallest queue depth. Thus, one difficulty of setting a server execution throttle according to this technique is the likelihood that many storage controllers on the SAN will operate at less than their maximum capacity.

Another difficulty of establishing the server execution throttles of the servers of the SAN concerns the process of individually or manually adjusting the execution throttles of one or more of the servers on the SAN. This task is often logistically challenging, burdensome, and time-consuming. An administrator of the SAN must calculate a new value for the execution throttle and must manually reset the execution throttle of every server on the SAN. The problem of resetting server execution throttle is exacerbated when a SAN has multiple storage devices that have different queue depths. Therefore, a method is needed to calculate and reset the execution throttle of servers on the SAN in a manner that is less burdensome and time-consuming to the administrator of the SAN. Another issue with the current method of managing the execution throttle of the servers on the SAN relates to the manual process that the administrator must perform to calculate and set the execution throttle of all the servers on the SAN. Because the process is manual in nature, relying primarily on hand calculations and the manual setting of the execution throttle of each server on the SAN, the process is subject to errors that are potentially very difficult to troubleshoot.

SUMMARY

In accordance with the present disclosure, a method and system is provided for correlating the execution throttle levels of the servers of a storage area network to the command queue depth of the storage controllers of the storage area network. The method described in the present disclosure involves determining for each storage controller of the network the logical storage units that are managed by the storage controller. For each storage controller, the servers are identified that have logical ownership of the logical storage units managed by each respective storage controller. For each storage controller, the execution throttle levels for those servers having logical ownership over logical storage units of the storage controller are summed, and the result is compared to the command queue depth of the respective storage controller. If, for any of the storage controllers of the network, the summed execution throttle level exceeds the command queue depth, the execution throttle level of one or more of the servers of the network must be adjusted to insure that the potential command throughput of the servers of the network is not greater than the maximum command throughput of the storage controllers of the network.

The method and system disclosed herein are advantageous in that they allow for the automated verification and adjustment of the execution throttle levels of the servers of the network. Because the execution throttle levels may be adjusted by an automated technique, it can be determined through an automated technique whether the entire network complies with the rule correlating the command throughput of the servers and storage controllers of the network. Another advantage of the method and network disclosed herein is the ability to verify whether a change to the execution throttle level of one server in the network has a detrimental effect on any of the other servers in the network. The method of the present disclosure accommodates the manual adjustment of execution throttle levels, while providing an automated means for verifying the throughput correlation of the network and automatically adjusting the execution throttle levels of one or more servers, if necessary.

The method and system disclosed herein are also advantageous in that the determination of whether the throughput correlation of the network is satisfied is made by examining the command queue depth of each storage controller, rather than relying on the command queue

depth of the storage controller having the shallowest command queue depth. In this manner, the execution throttle levels of the servers can be set so that the levels correspond more closely to the maximum throughput levels of the servers. In addition, a server can maintain a separate execution throttle for each storage device that the server accesses, which allows an execution throttle level to be made specific and managed independently for each storage controller, resulting in increased server performance.

Other technical advantages will be apparent to those of ordinary skill in the art in view of the following specification, claims, and drawings.

09760507.012601

BRIEF DESCRIPTION OF THE DRAWINGS

A more complete understanding of the present embodiments and advantages thereof may be acquired by referring to the following description taken in conjunction with the accompanying drawings, in which like reference numbers indicate like features, and wherein:

Figure 1 is a block diagram of an embodiment of a storage area network;

Figure 2 is a flow chart of a method for verifying and matching storage device queue depth to server execution throttle;

Figure 3 is a flow chart of a method for adjusting execution throttle;

Figure 4 is a flow chart of a method for monitoring command throughput of servers and adjusting execution throttle to accommodate server throughput levels; and

Figure 5 is a flow chart of a method for verifying and matching storage device queue depth to independent server execution throttle made specific to each storage controller.

09902701.012501

DETAILED DESCRIPTION

The present disclosure concerns a method and system for matching or correlating in a SAN the execution throttle of the servers of the network to the queue depth of the storage controllers of the network. The method described herein involves adjusting the execution throttle of the servers of the network as part of the determination of a suitable execution throttle level for each server of the network. A suitable execution throttle level is determined following an evaluation at each storage controller of the command queue depth of the storage controller and the logical ownership of the LUNs of the storage device by the servers of the network. The sum of the execution throttle or command throughput of those servers having logical ownership over a LUN managed by a single storage controller must not exceed the command queue depth or command throughput of that storage controller. By applying this rule to each of the storage controllers of the network, the execution throttle of each network server can be set so that no storage controller in the network will suffer from dropped commands or busy signals. In addition to the verification and adjustment algorithms outlined in the description herein, other automated verification and adjustment schemes can be employed to insure that potential command throughputs of the servers of the SAN and the maximum command throughputs of the storage controllers of the SAN are correlated on the basis of LUN ownership by servers of the SAN.

Shown in Figure 1 is a diagram of the architecture of a storage area network, which is indicated generally at 10. SAN 10 includes a number of servers 11, network switches 12, and storage devices 13. The physical storage within any storage device may be logically subdivided into one or more logical storage units. Each of these logical representations of physical storage is assigned a LUN. The term LUN is often used to refer both to the unique number that is assigned by the SAN to the logical storage unit and to the logical storage unit itself. In the network architecture of Figure 1, Storage Device X of Figure 1 includes a storage controller 14 and multiple logical storage units or LUNs 16. Each logical storage unit 16 is assigned a unique numeric identifier or LUN. For the sake of example, the LUNs of Figure 1 are shown as four digit binary numbers. Like Storage Device X, Storage Device Y includes a storage controller 14 and a number of logical storage units or LUNs 16, each having a unique logical identifier.

10507420 .012501

Each LUN 16 is logically normally owned by a single host server 11, although a LUN may be owned by multiple servers 11, as in the case of clustered servers. A single host server, however, may have logical ownership over multiple LUNs. As an example, Server A may have logical ownership over LUN 0001 in Storage Device X. Server B may have logical ownership over LUN 0010 in Server X and LUN 0100 in Storage Device Y. Server C may have logical ownership over LUN 0011 in Storage Device X and LUN 0101 in Storage Device Y. Server D may have logical ownership over LUN 0110 in Storage Device Y, and Server E may have logical ownership over LUN 0111 in Storage Device Y. As described, each LUN is assigned to just one network server, although a network server, such as Servers B and C in this example, may have logical ownership over multiple LUNs.

A flow diagram of the steps of setting the execution throttle of the host servers of the SAN is shown in Figure 2. At step 20, a LUN ownership map is retrieved. The LUN ownership map of the SAN identifies, for each LUN, the server or servers that have logical ownership over the LUN. In the example of Figure 1, each of servers A, B, and C has logical ownership over one of the LUNs in Storage Device X, and each of Servers B, C, D, and E has logical ownership over one of the LUNs in Storage Device Y. At step 22, the execution throttle levels or command throughput of each network server that has logical ownership over a LUN of the respective storage controller are summed. As part of step 22, a verification test is performed to determine whether the summed execution throttle value exceeds the command queue depth or command throughput of the associated storage controller. The rule of step 22 for this example is set out below in Equations 1 and 2:

Execution Throttle on Servers (A + B + C) < Command Queue Depth of Storage Device X — Equation 1

Execution Throttle on Servers (B + C + D + E) < Command Queue Depth of Storage Device Y — Equation 2

As to Equation 1, because each of Servers A, B, and C has logical ownership of a LUN on Storage Device X, the execution throttle of each of Servers A, B, and C is summed and compared to the command queue depth of Storage Device X. As to Equation 2, because each of Servers B, C, D, and E has logical ownership of a LUN on Storage Device Y, the execution throttle levels of each of Servers B, C, D, and E are summed and compared to the command queue depth of Storage Device Y.

The verification step of step 22 is performed for each storage controller of the SAN. If the verification step is not satisfied for any storage controller, processing continues at step 26, where the execution throttle is incremented or decremented by at least one increment on at least one of the servers associated with the storage controller associated with the unsatisfied verification step. At step 26, any number of algorithms may be applied to determine which of several potential servers should be adjusted and the amount by which the execution throttle of any server should be adjusted. As one example, the server associated with the unverified storage controller that has the highest execution throttle or throughput rate may be decremented by a set increment. As another example, a server whose execution throttle is adjusted is selected from the servers associated with the unverified storage controller in round robin fashion. The execution throttle adjustment step of step 26 is applied to the network with respect to each storage controller that did not pass the verification test of step 22. Once the adjustment step is complete, processing continues with the verification step of step 24. Step 22 (test), step 24 (verification), and step 26 (adjustment) iterate until the verification step is satisfied for each storage controller, i.e., the sum of the execution throttle levels of the servers having logical ownership over a LUN in a storage controller are not greater than the command queue depth of the storage controller, insuring that the commands sent from the servers to the storage controllers of the network will be handled by the storage controllers and will not encounter a busy signal or be dropped.

If the verification step of step 24 is satisfied for each storage controller, the execution throttle level of each server in the SAN is at an acceptable level, and the iterative adjustment and verification steps are complete. It is next determined at step 28 whether the execution throttle of

each server exceeds a minimum execution throttle setting. The logical rule of step 28 is set out in Equation 3:

Execution Throttle of Server A > Minimum Throttle Level Equation 3

It is possible to satisfy Equations 1 and 2 by intentionally setting the throttle levels of each of the servers to a low level, thereby insuring that instructions sent by the servers to the storage controllers do not overburden the command queues of the storage controllers. Doing so, however, would severely degrade the performance of the servers. Performing the minimum throttle level verification of step 28 insures that the servers will continue to be functional at the set throttle level. Enforcing a minimum execution throttle level also limits the number of unique servers that may have logical ownership over the LUNs managed by a storage controller. If the minimum throttle level verification of step 28 is satisfied, processing is complete. If the minimum throttle level verification of step 28 is not satisfied, an error message is provided to the network administrator that the automated throttle execution level process has failed. The steps in the flow diagram of Figure 2 are just one example of the steps or algorithm that can be performed to automate the process of adjusting the execution throttles of the servers of the SAN to insure that potential command queue depths of the storage controllers of the SAN are greater than or equal to the maximum potential command throughput of the servers of the SAN.

Aside from the automated process of setting the execution throttles of the servers of a SAN of Figure 2, a network administrator may manually set the execution throttles of the servers of the SAN. This manual setting of execution throttles may occur, for example, after the conclusion of or in conjunction with the automated assignment of execution throttles. As an example, and as shown in Figure 3, the administrator may at step 40 manually set the execution throttle of each server to its maximum value. At step 42, the automated process of adjusting the execution throttle levels of the servers of the SAN, such as the process described with respect to Figure 2 or 5, is initiated. The process of step 42 results in the automated incrementing or decrementing of the execution throttles of the servers of the network until it is verified that the command throughput of each server

of the SAN is set such that it is not possible for the command throughput of a server or the combined command throughput of a set of servers to exceed the command throughput of an associated storage controller of the network. At step 44, the result of the automated process of setting execution throttles for the servers of the SAN is a set of execution throttle settings that are correlated to the maximum command throughput of the associated storage controllers of the network. Thus, even though the administrator can manually set the execution throttle of one or more of the servers of the network, the automated throttle-setting process of the present disclosure, including the process steps described with respect to Figures 2 and 5, could be run to insure that the reset execution throttle levels of the affected servers are not set to levels that raise the possibility of dropped commands caused by an insufficient command queue depth on one or more of the storage controllers of the network.

A server can maintain a separate execution throttle for each storage device that is accessible by the server. As an example, a server may have access to LUNs that are managed by more than one storage controller. In this environment, the server may have a separate execution throttle level for each storage controller. This feature is useful in that it allows the network to more finely tune the execution throttle levels of the servers on a storage controller-specific basis in response to actual throughput of the server directed to individual storage controllers allowing for higher performance. For example, when the execution throttle of a server in relation to a particular storage controller is adjusted in order to optimize the throughput of the server in relation to the storage controller, it is not necessary to change the execution throttle of the server for all other storage controllers to which the server has access. Figure 5 shows a process for adjusting execution throttle that is similar to the process described with respect to Figure 2. The adjustment process of Figure 5 differs from that of Figure 2 in that at step 66 the server execution throttle is adjusted and managed independently for each storage controller. Thus, for each storage controller, the execution throttle levels of the servers having logical ownership of LUNs managed by the storage controller are set only with respect to the command throughput of each respective storage controller. The adjustment at step 66 allows a server to maintain a separate execution throttle for each storage device

that the server accesses and allows the execution throttle level to be made specific and managed independently for each storage controller.

The execution throttle levels of the servers of the network may also be adjusted in response to the operating characteristics of the network. An automated process may monitor the actual I/O demand of each server to identify those servers having higher command requests. This process may also monitor the input/output profile of each storage controller in the network to identify the most active servers of the group of servers that have logical ownership of the LUNs of each storage controller. As shown in Figure 4, the actual execution throughput of each server is monitored at step 50. At step 50, the input/output profile of each server is monitored in an attempt to detect an opportunity to adjust the execution throttle levels of the servers to account for I/O demand differences among the set of servers associated with a single storage controller. If it is determined at step 52 that there is a significant difference in the input/output profile among one or more of the set of servers associated with a single storage controller, processing continues at step 54, where the execution throttle of the more active server or servers of the set is incremented and the execution throttle of the less active server or servers of the set is decremented. The combination of steps 52 and 54 permits the network to identify when there is an opportunity to tune the performance of the servers in response to the actual throughput of the servers and to adjust the execution throttle of those servers in response to the actual throughput. Following this adjustment, the automated throttle adjustment process, such as the one described with respect to Figure 2 or 5, is initiated at step 56 to confirm that any adjustments made to the execution throttle levels of the servers of the network do not affect the operational integrity of the storage controllers of the network. At step 56, the execution throttle levels of the servers may be reset, if necessary, to correlate the command throughputs of the servers and storage controllers of the network. Following step 56, processing next continues at step 50. If it is determined at step 52 that there is not a significant disparity in the input/output profiles of the set of servers assigned to a certain storage controller, processing continues with the monitoring step of step 50 with the continued monitoring of the input/output profiles of the servers of the SAN.

The methodology disclosed herein for setting the execution throttles of the servers of a SAN is advantageous in that it takes into account the command queue depth of each storage controller when determining the setting of the execution throttle levels of the servers of the network. Rather than relying on the shallowest command queue depth as a guide or baseline for setting the execution throttle levels on the servers of the network, the technique disclosed herein sets the execution throttle levels of each server more precisely following an analysis of the potential throughput demands that could be placed on the storage controllers of the network that are associated with the respective servers of the network. Thus, rather than relying on a process for setting execution throttle levels that relies only on the storage controller having the shallowest command queue depth, the technique disclosed herein takes into account the networked or logical relationship between the servers, storage controllers, and logically owned storage units of the network.

The method and system disclosed herein also provide the opportunity to dynamically tune the throttle levels of the servers in response to the actual throughput characteristics of the network. The technique disclosed herein is also advantageous in that it can be automated and can be used to both accommodate and verify the manual adjustment of the execution throttle levels of the servers of the network. Further, the technique disclosed herein may be used to correlate the throughput of any related devices of a network and should not be limited in its application to a networked relationship of servers and storage controllers in a SAN. Rather, the methods of the present disclosure may be applied in any network relationship or configuration in which throughput limitations exist on related elements of the network.

Although the present disclosure has been described in detail, it should be understood that various changes, substitutions, and alterations can be made hereto without departing from the spirit and the scope of the invention as defined by the appended claims.